PartMC-HPC

Release 0.0.0

Zhonghua Zheng

Feb 27, 2023

ENVIRONMENTS SETUP

This website provides the information about how to install software (including compilers) relevant to PartMC, and perform analysis (e.g., using Jupyter) on HPC.

Author: Dr. Zhonghua Zheng

CHAPTER

ONE

CONDA INSTALLATION

Zhonghua Zheng (zhonghua.zheng@outlook.com)

Last update: 2021/04/26

1.1 Introduction

Here we would install Conda and create our first conda environment "partmc"

1.2 download and activate conda

```
# Download and install conda
$ cd $HOME
# If you are using Campus Cluster
$ cd /projects/your_path
# note: on Campus Cluster, we need to install under the /projects directory
# e.g., cd /projects/ctessum/zzheng25
$ wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh
$ chmod +x Miniconda3-latest-Linux-x86_64.sh
$ ./Miniconda3-latest-Linux-x86_64.sh

###### important ######
# Edit .bash_profile or .bashrc, add ":$HOME/miniconda3/bin"
# assume the original one is "PATH=$PATH:$HOME/bin"
# then we would have
# "PATH=$PATH:$HOME/bin:$HOME/miniconda3/bin"
# if we use Campus Cluster, please change it accordingly, e.g.,
# "PATH=$PATH:$HOME/bin:/projects/ctessum/zzheng25/miniconda3/bin"
########################

# Activate the conda system, depends on which one you edited
$ source .bash_profile
$ source .bashrc
```

1.3 create and activate a conda environment

1.3.1 option 1: install the environment manually

```
# assume we want to create an environment named "partmc", with Python version 3.7

# we only need to create the environment once
$ conda create -n partmc python=3.7

# activate this conda environment
# we would do this everytime
$ source activate partmc

# assume we want to install some useful Python packages, e.g., numpy, pandas, scipy,␣
↪scikit-learn, netcdf4, xarray, matplotlib, jupyterlab
$ conda install -c conda-forge numpy pandas scipy scikit-learn netcdf4 xarray matplotlib␣
↪jupyterlab
# assume we want to install something that is not available in conda-forge
$ pip install xgboost==1.1.1
```

1.3.2 option 2: install the environment using a file

create a file, rename it to be "environment.yml"

```
channels:
- conda-forge
- defaults

dependencies:
- python=3.7.0
- numpy
- pandas
- scipy
- scikit-learn
- netcdf4
- xarray
- matplotlib
- jupyterlab
- ipykernel
- ipywidgets
- pip
- pip:
  - xgboost==1.1.1
name: partmc
```

then run the commands

```
# Create an environment "partmc" and install the necessary packages
$ conda env create -f environment.yml
# Activate the "partmc" environment
$ source activate partmc
```

CHAPTER

TWO

SPACK: HPC PACKAGES AND COMPLIERS INSTALLATION

Zhonghua Zheng (zhonghua.zheng@outlook.com)

Last update: 2021/04/26

2.1 Introduction

Below is an example of installing “gcc”, “hdf5”, “netcdf”, “cmake”

2.2 Prerequisites

Make sure you have “git” and “Python” available.

If “Python” is not available, please follow the “**Conda Installation**”

2.3 option 1: install from a script

```
$ mkdir /projects/your_path
$ module load git
$ module load python
# if you have followed the "Conda Installation"
# $ source activate partmc
$ git clone https://github.com/spack/spack.git
```

2.3.1 1.1 create a script

please modify the following commands appropriately with your partition, netid and path

- `#SBATCH --partition=xxx` -> `#SBATCH --partition=ctessum`
- `#SBATCH --mail-user=your_netid@illinois.edu`
- `export SPACK_ROOT=/projects/your_path/spack`

```
#!/bin/bash

#SBATCH --partition=xxx
```

(continues on next page)

(continued from previous page)

```
#SBATCH --nodes=1
#SBATCH --mem=64g
#SBATCH --time=12:00:00
#SBATCH --job-name=spack_install
#SBATCH --mail-type=ALL
#SBATCH --mail-user=your_netid@illinois.edu

export SPACK_ROOT=/projects/your_path/spack
source ${SPACK_ROOT}/share/spack/setup-env.sh
spack install gcc@9.3.0 %gcc@4.8.5 target=x86_64
spack load gcc@9.3.0
spack compiler find
spack install hdf5%gcc@9.3.0 target=x86_64 +cxx+fortran+hl+pic+shared+threadsafe
spack load hdf5%gcc@9.3.0
spack install netcdf-fortran%gcc@9.3.0 target=x86_64 ^
↪hdf5+cxx+fortran+hl+pic+shared+threadsafe
spack load netcdf-fortran
spack load netcdf-c
spack install cmake%gcc@9.3.0 target=x86_64
```

2.3.2 1.2 Run the command

assume the name of the script is "install_spack.sh"

```
$ sbatch install_spack.sh
```

- took ~8 hours

2.4 option 2: install it manually

Please use the **compute note** (recommand 8 hrs):

```
$ srun --partition=xxx --nodes=1 --mem=64g  --time=08:00:00 --pty bash -i
# e.g., srun --partition=ctessum --nodes=1 --mem=64g  --time=08:00:00 --pty bash -i
$ mkdir /projects/your_path
$ module load git
$ module load python
# if you have followed the "Conda Installation"
# $ source activate partmc
$ git clone https://github.com/spack/spack.git
$ export SPACK_ROOT=/projects/your_path/spack
$ source ${SPACK_ROOT}/share/spack/setup-env.sh
# use gcc@4.8.5 to build gcc@9.3.0
$ spack install gcc@9.3.0 %gcc@4.8.5 target=x86_64
$ spack load gcc@9.3.0
$ spack compiler find
$ spack install hdf5%gcc@9.3.0 target=x86_64 +cxx+fortran+hl+pic+shared+threadsafe
$ spack load hdf5%gcc@9.3.0
$ spack install netcdf-fortran%gcc@9.3.0 target=x86_64 ^
```

(continues on next page)

(continued from previous page)

```
↪hdf5+cxx+fortran+hl+pic+shared+threadsafe
$ spack load netcdf-fortran
$ spack load netcdf-c
$ spack install cmake%gcc@9.3.0 target=x86_64
```

2.5 useful commands and scripts

commands

```
# find location
$ spack location -i gcc@9.3.0
# add new compiler
$ spack compiler find
# Spack compilers should print out a list of available compilers
$ spack compilers
# Spack will print out a long list of info.
$ spack config get compilers
```

scripts

Below is the script to load the spack environment. Please lease change the path accordingly

```
#!/bin/bash

export SPACK_ROOT=/projects/ctessum/zzheng25/spack
source ${SPACK_ROOT}/share/spack/setup-env.sh
spack load gcc@9.3.0
spack compiler find
spack compilers
spack config get compilers
spack load hdf5%gcc@9.3.0
spack load netcdf-fortran
spack load netcdf-c
spack load cmake
export CC=gcc
export FC=gfortran
which gcc
```

2.6 Reference

GEOS-Chem: tutorial, GitHub discussion (for full installation including cdo)

Chinese: link

CHAPTER

THREE

PYTHON AND JUPYTER

Zhonghua Zheng (zhonghua.zheng@outlook.com)

Last update: 2022/02/17

3.1 Introduction

Here we would

- use Jupyter notebook on HPC with a GPU

3.2 Prerequisites

Make sure you have a conda environment ("partmc") available.

If the "partmc" conda environment is not available, please follow the "**Conda Installation**"

3.3 use Jupyter notebook on HPC with a GPU

step 1: run the following script

Please change the partition and gpu configuration accordingly

```
#!/bin/bash

# if use gpu:
srun --partition=ctessum --nodes=1 --time=03:00:00 --gres=gpu:QuadroRTX6000:1 --pty bash␣
↪-i
# cpu only:
# srun --partition=ctessum --nodes=1 --time=03:00:00 --pty bash -i
source activate partmc
echo "ssh -N -L 8880:`hostname -i`:8880 $USER@cc-login.campuscluster.illinois.edu"
jupyter notebook --port=8880 --no-browser --ip=`hostname -i`
```

Note: if the command from "echo" doesn't work. Please use the command below as a replacement

```
echo "ssh -t -t $USER@cc-login.campuscluster.illinois.edu -L 8880:localhost:8880 ssh␣
↪`hostname` -L 8880:localhost:8880"

# a reference for UCAR's HPC
echo "ssh -N -L 8880:`hostname`:8880 $USER@`hostname`.ucar.edu"
jupyter notebook --port=8880 --no-browser --ip=`hostname`
```

Note: if you are using **keeling**:

```
qsub -I -l select=1:ncpus=32 -l walltime=24:00:00
source activate
conda activate partmc
echo "ssh -N -L 8880:127.0.0.1:8880 $USER@keeling.earth.illinois.edu"
jupyter notebook --port=8880 --no-browser --ip=127.0.0.1
```

step 2: launch a new terminal, copy and paste the command printed by the "echo" command, and log in

step 3: open your browse (e.g., Google Chrome), type `https://localhost:8880`

3.4 Trouble Shooting

GPU relevant command

```
$ lspci | grep -i nvidia
$ nvidia-smi
```

kill session:

```
$ ps -u your_netid -f | grep ssh
$ kill -9 session_id
```

for nfs:

```
$ lsof | grep nfs00000
$ kill -9 session_id
```

CHAPTER

FOUR

JULIA AND JUPYTER

Zhonghua Zheng (zhonghua.zheng@outlook.com)

Last update: 2021/04/26

4.1 Introduction

Here we would

- download and install Julia
- create our conda environment "julia"
- use Jupyter notebook on HPC with a GPU

4.2 Prerequisites

Make sure you have "git" and "conda" available.

If "conda" is not available, please follow the "**Conda Installation**"

4.3 download and Install Julia

Note here we create another environment "julia" other than "partmc".

```
# Download and install julia: https://julialang.org/downloads/platform/#linux_and_freebsd
$ cd $HOME
# If you are using Campus Cluster
$ cd /projects/your_path
$ wget https://julialang-s3.julialang.org/bin/linux/x64/1.6/julia-1.6.1-linux-x86_64.tar.
↪gz
$ tar zxvf julia-1.6.1-linux-x86_64.tar.gz
$ mv julia-1.6.1 julia

###### important ######
# Edit .bash_profile or .bashrc, add ":$HOME/julia/bin"
# assume the original one is "PATH=$PATH:$HOME/bin:$HOME/miniconda3/bin"
# then we would have
# "PATH=$PATH:$HOME/bin:$HOME/miniconda3/bin:$HOME/julia/bin"
```

(continues on next page)

(continued from previous page)

```
# if we use Campus Cluster, please change it accordingly, e.g.,
# "PATH=$PATH:$HOME/bin:/projects/ctessum/zzheng25/miniconda3/bin:/projects/ctessum/
↪zzheng25/julia/bin"
########################

# Activate the conda system, depends on which one you edited
$ source .bash_profile
$ source .bashrc
```

4.4 create our conda environment “julia”

```
# create conda environment and install jupyter notebook
$ conda create -n julia
$ conda activate julia
$ conda install -c conda-forge python=3.7 jupyter

# ref: https://github.com/JuliaLang/IJulia.jl
$ julia
# please change the path accordingly
julia> ENV["JUPYTER"] ="/projects/ctessum/zzheng25/miniconda3/envs/julia/bin/jupyter"
julia> using Pkg; Pkg.instantiate()
julia> Pkg.add("JLLWrappers"); Pkg.add("libsodium_jll"); Pkg.add("ZMQ")
# julia> Pkg.build("libsodium_jll"); Pkg.build("ZMQ")
julia> using JLLWrappers, libsodium_jll, ZMQ
julia> Pkg.add("IJulia")
julia> using IJulia
julia> Pkg.add("CUDA")
julia> using CUDA

$ ln -s your_julia_location/bin/julia your_conda_location/envs/julia/bin/julia
# e.g., $ ln -s /projects/ctessum/zzheng25/julia/bin/julia /projects/ctessum/zzheng25/
↪miniconda3/envs/julia/bin/julia
```

4.5 use Jupyter notebook on HPC with a GPU

step 1: run the following script

```
#!/bin/bash

# if use gpu:
srun --partition=ctessum --nodes=1 --time=03:00:00 --gres=gpu:QuadroRTX6000:1 --pty bash␣
↪-i
# cpu only:
# srun --partition=ctessum --nodes=1 --time=03:00:00 --pty bash -i
source activate julia
echo "ssh -N -L 8880:`hostname -i`:8880 $USER@cc-login.campuscluster.illinois.edu"
jupyter notebook --port=8880 --no-browser --ip=`hostname -i`
```

step 2: launch a new terminal, copy and paste the command printed by the "echo" command, and log in

step 3: open your browse (e.g., Google Chrome), type `https://localhost:8880`

4.6 Trouble Shooting

GPU relevant commands

```
$ lspci | grep -i nvidia
$ nvidia-smi
julia > CUDA.version()
julia > CUDA.versioninfo()
julia > [CUDA.capability(dev) for dev in CUDA.devices()]
```

kill session:

```
$ ps -u your_netid -f | grep ssh
$ kill -9 session_id
```

for nfs:

```
$ lsof | grep nfs00000
$ kill -9 session_id
```

CHAPTER

FIVE

PARTMC-MOSAIC INSTALLATION

Zhonghua Zheng (zhonghua.zheng@outlook.com)

Last update: 2021/04/26

5.1 Prerequisites

Make sure you have "git" and "spack" available.

- If "spack" is not available, please follow the "**spack: HPC Packages and Compliers Installation**"

Make sure you to have the permission to use the MOSAIC software

5.2 load spack environment

run the following script, please use your own SPACK_ROOT path

```
#!/bin/bash

export SPACK_ROOT=/projects/ctessum/zzheng25/spack
source ${SPACK_ROOT}/share/spack/setup-env.sh
spack load gcc@9.3.0
spack compiler find
spack compilers
spack config get compilers
spack load hdf5%gcc@9.3.0
spack load netcdf-fortran
spack load netcdf-c
spack load cmake
export CC=gcc
export FC=gfortran
which gcc
```

5.3 build MOSAIC chemistry

```
$ cd /projects/your_path
$ tar -zxvf mosaic-2012-01-25.tar.gz
$ mv mosaic-2012-01-25 mosaic
$ cd /projects/your_path/mosaic
$ cp Makefile.local.gfortran Makefile.local
$ make
```

5.4 build PartMC-MOSAIC

step 1: use the following commands to get the paths for later use (step 3)

```
$ spack location -i netcdf-fortran
$ spack location -i netcdf-c
$ spack location -i cmake
```

step 2: run the following commands

```
$ cd /projects/your_path
$ module load git
$ git clone git@github.com:compdyn/partmc.git
$ cd partmc
$ mkdir build
$ cd build
$ export MOSAIC_HOME=/projects/your_path/mosaic
$ ccmake ..
```

step 3: First, press “c”. Then press “e”, and type the following options (use the paths you got from step 1):

```
CMAKE_BUILD_TYPE: RELEASE

cmake_install_PREFIX:
/projects/your_path/spack/opt/spack/linux-centos7-x86_64/gcc-9.3.0/cmake-XXXXXX/bin

ENABLE_MOSAIC: ON

NETCDF_C_LIB:
/projects/ctessum/zzheng25/spack/opt/spack/linux-centos7-x86_64/gcc-9.3.0/netcdf-c-
↪XXXXXX/lib/libnetcdf.a

NETCDF_FORTRAN_LIB:
/projects/ctessum/zzheng25/spack/opt/spack/linux-centos7-x86_64/gcc-9.3.0/netcdf-fortran-
↪XXXXXX/lib/libnetcdff.a

NETCDF_INCLUDE_DIR:
/projects/ctessum/zzheng25/spack/opt/spack/linux-centos7-x86_64/gcc-9.3.0/netcdf-fortran-
↪XXXXXX/include
```

step 4: press “c”, then “c” again, and “g”

step 5: compile and make sure you have all the test cases such as “test_mosaic_1” and “test_mosaic_2” passed.

```
make
make test
```

CHAPTER

SIX

PARTMC-MOSAIC INSTALLATION ON KEELING

Zhonghua Zheng and Jeffrey Curtis (jcurtis2 at illinois.edu)

Last update: 2021/05/30

6.1 Prerequisites

Make sure you have access to keeling

Make sure you to have the permission to use the MOSAIC software

6.2 access keeling and prepare install

```
# from your local machine
$ ssh -y your_NetID@keeling7.earth.illinois.edu
# launch an interactive job
$ qsub -I -l select=1:ncpus=32 -l walltime=24:00:00
$ module load gnu/gnu-9.3.0
$ module load gnu/netcdf4-4.7.4-gnu-9.3.0
$ module load gnu/openmpi-3.1.6-gnu-9.3.0
$ export FC=gfortran
$ export CC=gcc
```

You can check if you have loaded successfully by typing

```
$ module list
```

You can check if you have set up FC and CC successfully by typing

```
$ which $FC
$ which $CC
```

6.3 build MOSAIC chemistry

```
$ cd your_path
$ tar -zxvf mosaic-2012-01-25.tar.gz
$ mv mosaic-2012-01-25 mosaic
$ cd your_path/mosaic
$ cp Makefile.local.gfortran Makefile.local
$ make
```

6.4 build PartMC-MOSAIC

step 1: download PartMC and set up

```
$ cd your_path
$ module load git # you may skip this if the command below works
$ git clone git@github.com:compdyn/partmc.git
$ cd partmc
$ mkdir build
$ cd build
$ export NETCDF_HOME=/sw/netcdf4-4.7.4-gnu-9.3.0
$ export MOSAIC_HOME=your_path/mosaic
$ ccmake ..
```

step 2: First, press "c". Then press "e", and type the following options

```
CMAKE_BUILD_TYPE: RELEASE

ENABLE_MOSAIC: ON

NETCDF_C_LIB:
/sw/netcdf4-4.7.4-gnu-9.3.0/lib/libnetcdf.so

NETCDF_FORTRAN_LIB:
/sw/netcdf4-4.7.4-gnu-9.3.0/lib/libnetcdff.so

NETCDF_INCLUDE_DIR:
/sw/netcdf4-4.7.4-gnu-9.3.0/include
```

step 3: press "c", then "c" again, and "g"

step 4: compile and make sure you have all the test cases such as "test_mosaic_1" and "test_mosaic_2" passed.

```
make
make test
```

CHAPTER

SEVEN

PARTMC-MOSAIC-MCM INSTALLATION

Zhonghua Zheng (zhonghua.zheng@outlook.com) and Xiaokai Yang

Last update: 2021/04/26

7.1 Prerequisites

Make sure you have "git" and "spack" available.

- If "spack" is not available, please follow the "**spack: HPC Packages and Compliers Installation**"

Make sure you to have the permission to use the MOSAIC software

7.2 load spack environment

run the following script, please use your own SPACK_ROOT path

```
#!/bin/bash

export SPACK_ROOT=/projects/ctessum/zzheng25/spack
source ${SPACK_ROOT}/share/spack/setup-env.sh
spack load gcc@9.3.0
spack compiler find
spack compilers
spack config get compilers
spack load hdf5%gcc@9.3.0
spack load netcdf-fortran
spack load netcdf-c
spack load cmake
export CC=gcc
export FC=gfortran
which gcc
```

7.3 download MOSAIC-MCM

run the following command

```
$ cd /projects/your_path
$ tar -zxvf mosaic-2012-01-25.tar.gz
$ tar -zxvf PartMC-MOSAIC-MCMv3.3.1.tar.gz
$ mv PartMC-MOSAIC-MCMv3.3.1 partmc-mcm
```

7.4 build MOSAIC-MCM

Create a script `mcm_compile.sh` to compile MOSAIC-MCM.

Modify your `partition`, `job-name`, and `mail-user` properly.

```
#!/bin/bash

#SBATCH --partition=ctessum
#SBATCH --nodes=1
#SBATCH --mem=64g
#SBATCH --time=2:00:00
#SBATCH --job-name=mcm_compile
#SBATCH --mail-type=ALL
#SBATCH --mail-user=xxxxxx@illinois.edu

cd /projects/ctessum/zzheng25/partmc-mcm/MOSAIC-MCM
make
```

Submit your script. It would take around 0.5 hrs.

```
sbatch mcm_compile.sh
```

7.5 build PartMC-MOSAIC-MCM

step 1: use the following commands to get the paths for later use (step 3)

```
$ spack location -i netcdf-fortran
$ spack location -i netcdf-c
$ spack location -i cmake
```

step 2: run the following commands

```
$ cd /projects/your_path
$ module load git
$ git clone git@github.com:compdyn/partmc.git
######## important ########
# cd /projects/your_path/partmc/src
# modify the lines, have 1000 -> 10000
# call assert(719237193, n_spec < 1000)
# character(len=((GAS_NAME_LEN + 2) * 1000)) :: gas_species_names
```

(continues on next page)

(continued from previous page)

```
############################
$ cd /projects/your_path/partmc/
$ mkdir build
$ cd build
$ ccmake ..
```

step 3: First, press “c”. Then press “e”, and type the following options (use the paths you got from step 1):

```
CMAKE_BUILD_TYPE: RELEASE

cmake_install_PREFIX:
/projects/your_path/spack/opt/spack/linux-centos7-x86_64/gcc-9.3.0/cmake-XXXXXX/bin

ENABLE_MOSAIC: ON

NETCDF_C_LIB:
/projects/ctessum/zzheng25/spack/opt/spack/linux-centos7-x86_64/gcc-9.3.0/netcdf-c-
↪XXXXXX/lib/libnetcdf.a

NETCDF_FORTRAN_LIB:
/projects/ctessum/zzheng25/spack/opt/spack/linux-centos7-x86_64/gcc-9.3.0/netcdf-fortran-
↪XXXXXX/lib/libnetcdff.a

NETCDF_INCLUDE_DIR:
/projects/ctessum/zzheng25/spack/opt/spack/linux-centos7-x86_64/gcc-9.3.0/netcdf-fortran-
↪XXXXXX/include

MOSAIC_INCLUDE_DIR:
/projects/ctessum/zzheng25/partmc-mcm/MOSAIC-MCM/datamodules

MOSAIC_LIB:
/projects/ctessum/zzheng25/partmc-mcm/MOSAIC-MCM/libmosaic.a
```

step 4: press “c”, then “c” again, and “g”

step 5: compile and make sure you have all the test cases passed, except for “test 48” and “test 50”

```
make
make test
```

CHAPTER

EIGHT

PARTMC-MOSAIC-MCM INSTALLATION ON KEELING

Zhonghua Zheng (zhonghua.zheng@outlook.com)

Last update: 2021/05/30

8.1 Prerequisites

Make sure you have access to

- keeling7
- github.com:xiaoky97/MCM-PartMC-MOSAIC

8.2 load environment

```
$ qsub -I -l select=1:ncpus=32 -l walltime=24:00:00
$ module load gnu/gnu-9.3.0
$ module load gnu/netcdf4-4.7.4-gnu-9.3.0
$ module load gnu/openmpi-3.1.6-gnu-9.3.0
$ export FC=gfortran
$ export CC=gcc
```

8.3 download MOSAIC-MCM

run the following command

```
$ cd ~
$ git clone git@github.com:xiaoky97/MCM-PartMC-MOSAIC.git
```

8.4 build MOSAIC-MCM

```
$ cd ~/MCM-PartMC-MOSAIC/MOSAIC-MCM
make
```

It takes ~30 mins.

8.5 build PartMC-MOSAIC-MCM

step 1: set up

```
$ export NETCDF_HOME=/sw/netcdf4-4.7.4-gnu-9.3.0
$ cd ~/MCM-PartMC-MOSAIC/PartMC
$ export NETCDF_HOME=/sw/netcdf4-4.7.4-gnu-9.3.0
$ mkdir build
$ cd build
$ ccmake ..
```

step 2: First, press "c". Then press "e", and type the following options (use the paths you got from step 1):

```
CMAKE_BUILD_TYPE: RELEASE

ENABLE_MOSAIC: ON

NETCDF_C_LIB:
/sw/netcdf4-4.7.4-gnu-9.3.0/lib/libnetcdf.so

NETCDF_FORTRAN_LIB:
/sw/netcdf4-4.7.4-gnu-9.3.0/lib/libnetcdff.so

NETCDF_INCLUDE_DIR:
/sw/netcdf4-4.7.4-gnu-9.3.0/include

MOSAIC_INCLUDE_DIR (using your MOSAIC-MCM path):
/data/keeling/a/zzheng25/MCM-PartMC-MOSAIC/MOSAIC-MCM/datamodules

MOSAIC_LIB (using your MOSAIC-MCM path):
/data/keeling/a/zzheng25/MCM-PartMC-MOSAIC/MOSAIC-MCM/libmosaic.a
```

step 3: press "c", then "c" again, and "g"

step 4: compile and make sure you have all the test cases passed, except for "test 48" and "test 50"

```
make
make test
```

CHAPTER

NINE

MULTIPLE SCENARIOS WITH SCHEDULER

Zhonghua Zheng (zhonghua.zheng@outlook.com)

Last update: 2021/05/23

Note: If you turn mosaic off, you can't have "do_optical" setup in the ".spec" file.

9.1 Introduction

Here we would run multiple scenarios using NCSA's Scheduler

We can name the script as "case_mcm_300.sh". This script would launch 300 PartMC scenarios in parallel.

Here we have to modify the script accordingly (See INSTRUCTION within the script), then submit the job by executing `sbatch case_mcm_300.sh`

```
#!/bin/bash
#SBATCH --job-name=case_mcm_300
#SBATCH -n 301
#SBATCH -p sesebig
#SBATCH --time=24:00:00
#SBATCH --mem-per-cpu=4000
# Email if failed run
#SBATCH --mail-type=FAIL
# Email when finished
#SBATCH --mail-type=END
# My email address
#SBATCH --mail-user=your_email@illinois.edu

# the parent_path contains cases/ folder
export SLURM_SUBMIT_DIR=/data/keeling/a/zzheng25/scenario_generator
# case folder name under cases/case_*
export case=case_mcm_300
# number of (scenarios+1)
export scenario_num_plus_1=301
# PartMC path
export PMC_PATH=/data/keeling/a/zzheng25/partmc-mcm/PartMC
# Path for results
export WORK_DIR=/data/keeling/a/zzheng25/d/mcm_test

## INSTRUCTION
# module load gnu/openmpi-3.1.6-gnu-9.3.0
```

(continues on next page)

(continued from previous page)

```
# cd Scheduler
# mpif90 -o scheduler.x scheduler.F90
# mv scheduler.x ..
# chmod 744 partmc_submit.sh
# sed -i 's/\r//g' partmc_submit.sh
#
# Within partmc_submit.sh:
# - define your job name, e.g., "SBATCH --job-name=case_mcm_300"
# - define "#SBATCH -n XXX" (XXX should be the same as "scenario_num_plus_1")
# - define the user email
# - define the variables in export XXX
#
# type: sbatch case_mcm_300.sh

######## Do not Change ########
# The job script can create its own job-ID-unique directory
# to run within.  In that case you'll need to create and populate that
# directory with executables and inputs
mkdir -p $WORK_DIR/$SLURM_JOB_ID
cd $WORK_DIR/$SLURM_JOB_ID
cp -r $PMC_PATH/build build
cp -r $PMC_PATH/src src

# Copy the scenario directory that holds all the inputs files
cp -r $SLURM_SUBMIT_DIR/cases/$case/scenarios .

# Copy things to run this job
# Need the scheduler and the joblist
cp $SLURM_SUBMIT_DIR/scheduler.x .
cp $SLURM_SUBMIT_DIR/cases/$case/joblist .

# Run the library. One core per job plus one for the master.
mpirun -np $scenario_num_plus_1 ./scheduler.x joblist /bin/bash -noexit -nostdout > log
```